

A New Compressive Video Sensing Framework for Mobile Broadcast

Chengbo Li, Hong Jiang, Paul Wilford, Yin Zhang, and Mike Scheutzow

Abstract—A new video coding method based on compressive sampling is proposed. In this method, a video is coded using compressive measurements on video cubes. Video reconstruction is performed by minimization of total variation (TV) of the pixelwise discrete cosine transform coefficients along the temporal direction. A new reconstruction algorithm is developed from TVAL3, an efficient TV minimization algorithm based on the alternating minimization and augmented Lagrangian methods. Video coding with this method is inherently scalable, and has applications in mobile broadcast.

Index Terms—Alternating minimization, augmented Lagrangian method, compressive sensing, discrete cosine transform (DCT), scalable video coding, total variation (TV), video coding.

I. INTRODUCTION

IN A MOBILE video broadcast network, a video source is broadcast to multiple clients having different characteristics. The clients may have different channel capacities, different display resolutions, or different computing resources. It is therefore desirable for a video source to be encoded in such a way that a single encoded video stream can be transmitted, and be optimally usable by all the diverse clients. In other words, we want to encode and transmit the video source once, but allow any subset of the bit stream to be successfully decoded by a client, possibly at different complexities.

The traditional video coding such as MPEG (MPEG2 or MPEG4) does not provide the scalability described above. The lack of scalability exhibits itself in at least two ways. First, an encoded video is not scalable with transmission channel capacity. Because of its fixed bit rate, an encoded stream is unusable in a channel which does not support the required bit rate, and is also suboptimal in a channel having higher bit rate. For MPEG coders, uncorrected transmission errors cause reduced video quality, and the rapid drop off in quality as error rate increases is known as the "cliff effect." Second, the MPEG video is not scalable with decoding complexity or decoded quality. An encoded video bitstream decodes only

one way and with a fixed complexity (not considering post-processing such as resizing, or enhancement, after decoding). It is necessary to create multiple encoded streams of the same video content to support decoders of different qualities.

Efforts have been made to introduce multi-resolution scalability into video coding, noticeably by the scalable video coding (SVC) feature of H.264 [1] and the wavelet transform of Motion JPEG 2000 [2]. Both methods encode video into ordered layers, or levels, of streams, and the resolution, or quality, of the decoded video increases progressively as higher layers are processed at the decoder. Hierarchical modulation [3] may be used in conjunction with these scalable video codes to achieve more bandwidth efficiency. For example, the high priority of hierarchical modulation can be used to carry lower layers of the encoded video, and low priority of hierarchical modulation can be used to carry higher layers of the encoded video.

There has been an abundance of research activity in video coding to provide scalable decoding resolution, see [4]–[7]. A joint video coding and transmission method was proposed in [8] to provide scalability with transmission channel capacity. These activities are a response to the fact that the scalability provided by H.264 or Motion JPEG 2000 is still not satisfactory. Specifically, the ordered layer structure does not provide scalability at a fundamental level, because a video encoded using these approaches must be decoded at the lowest layer, and progressively built up by higher layers. The loss of a lower layer in transmission makes the higher layer data useless, even when the higher layer data is received error-free. Therefore, the ordered layer structure is not scalable with the channel capacity [8]. Protecting an encoded (non-scalable) video stream using fountain codes does not solve this problem, because a fountain code only provides reliable delivery of the encoded video stream, but does not provide scalability to the video itself. Other error-resilient video transmission methods are considered in [24]–[25].

Based on early efforts on ℓ_1 minimization, Candès, Romberg, Tao [9], [19], and Donoho [20] developed compressive sensing (CS) theory which proves that a sparse signal under some sparsity basis can be precisely recovered using a small number of measurements. Due to the proliferation of compressive sampling techniques, video coding using compressive measurements is rapidly emerging [10]–[11]. Compressive video sensing offers the scalability desired in a video network [12]–[13], and is also suitable for lossy wireless transmission [14]. When measurements of a video are made using a random (or pseudo-random) matrix, the video source

Manuscript received December 22, 2011; revised August 8, 2012; accepted October 1, 2012. Date of current version February 20, 2013. This paper was recommended by Associate Editor C. Zhu.

C. Li and Y. Zhang are with the Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005 USA (e-mail: steven.lib@gmail.com; yzhang@rice.edu).

H. Jiang, P. Wilford, and M. Scheutzow are with Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ 07974 USA (e-mail: hong.jiang@alcatel-lucent.com; paul.wilford@alcatel-lucent.com; mike.scheutzow@alcatel-lucent.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TBC.2012.2226509

information is distributed among the measurements of equal significance. Particularly, this means that no one measurement is more important than another. The reconstruction of video requires a certain number of measurements to be available, but it does not require the availability of a particular measurement. In this sense, a measurement lost in transmission can simply be replaced by any other measurement. Furthermore, since a video does not have a well-defined sparsity, statistically, when more measurements are used in reconstruction, the image quality of the reconstructed video improves [15]. If the measurements of the video are transmitted by broadcast or multicast, a receiver in a higher capacity channel will have more measurements available, and hence a reconstructed video of higher quality, than a receiver in a lower capacity channel. These properties illustrate that video coding using compressive sampling is inherently scalable with the channel capacity, and does not suffer from the cliff effect in broadcast or multicast.

In this brief paper, we propose a framework for video coding using compressive measurements in which an encoded video is scalable both with the channel capacity and with decoding complexity. In this framework, a source video is divided into self-contained video cubes. A random matrix is used for making the measurements. The resulting measurements are indistinguishable in the sense that the quality of the reconstructed video improves as the number of measurements increase, but does not rely on any particular set of measurements. In the proposed framework, the reconstruction is performed by minimizing the total variation (TV) of the coefficients of the pixelwise discrete cosine transform (DCT) in the temporal direction. TV minimization was first introduced for image denoising problems by Rudin, Osher, and Fatemi [21], and applied to compressive sensing by Candès, Romberg, Tao [9]. In general, it succeeds when the gradient of the underlying signal or image is sparse. However, TV has been shown to be effective for image reconstruction on a much larger scale empirically. The discrete cosine transform has been widely used as the sparsity basis for compressive sensing due to its ability to remove the periodic redundancy in a signal. Therefore, the proposed TV-DCT method has the advantage of preserving the spatial edges of images using TV, and, at the same time, removing the temporal redundancy using the DCT.

Also in this brief paper, we extend the TVAL3 algorithm [17], originally developed for solving 2-D TV minimization problems, to the application of video reconstruction. A new algorithm based on TVAL3 is developed to solve the TV-DCT minimization problem of this brief paper.

The focus of this brief paper is on using compressive sensing as an effective coding method for video transmission in broadcast channels. In particular, we address the issue of how to avoid the cliff effect and how to encode the video so that a single encoded bit stream can be used in channels of different bandwidth, and by decoders of different resolution. In this brief paper, this goal is achieved by using compressive measurements on video cubes. During video reconstruction, the spatial total variation (TV) and temporal discrete cosine transform (DCT) are used as sparsifying operators.

The method of this brief paper is similar to the methods used in MRI imaging [27]–[28] in the sense that the signals, video

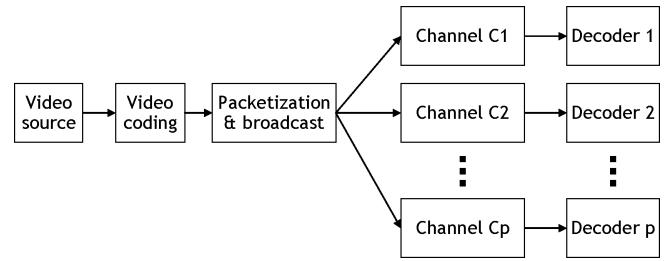


Fig. 1. Video coding for wireless transmission.

in this brief paper, and dynamic MRI images in [27]–[28], are sparse both spatially and temporally, and the sparsity can be exploited by using spatial and temporal transforms. The MRI methods, however, are different. In [27], the measurements are made in the Fourier transform domain while the measurements of video in this brief paper are made in the time domain. The method of [28] uses predictions and residuals similar to MPEG, so it suffers from the cliff effect.

The method of this brief paper is also different from that of [13] in two aspects. First, key frames are used in [13]. The use of key frames causes cliff effects because the loss of a key frame has a significant impact on the decoded video. Secondly, a feedback channel must be used in [13] while we are concerned with broadcast in which a feedback channel is typically not available.

This brief paper is organized as follows. The new framework for video coding is introduced in Section II. A new algorithm extended from TVAL3 is described in Section III. Numerical results are presented in Section IV. The conclusion and discussion are covered in Section V.

II. VIDEO CODING USING COMPRESSIVE SAMPLING

In this section, a framework is developed in which a source video is coded using compressive sampling, broadcast in a wireless network and reconstructed by multiple clients as shown in Fig. 1.

Our goal is to use an efficient video coding that is scalable in both capacity and decoding complexity. In this framework, the source video is first divided into video cubes. Then the measurements of each cube are generated by random linear combinations of pixels and transmitted. In the decoder, video reconstruction is performed by minimizing the 2-D TV of the time domain DCT coefficients for each video cube.

A. Video Coding Using Compressive Sampling

A source video consists of a number of frames with a resolution of $P \times Q$, where P and Q are the numbers of the horizontal and vertical pixels, respectively, for each frame of the video. To efficiently encode it, we process the source video block by block. First, the video source is divided into non-intersecting cubes, each of which consists of r frames of a sub-region of size $p \times q$. For simplicity of description, we assume that each frame of a video cube is taken from the same spatial region in its respective frame of the source video. A video cube is illustrated in Fig. 2 in which the video cube has 4 frames, and each is taken from a sub-region in the top right corner of a full video frame.

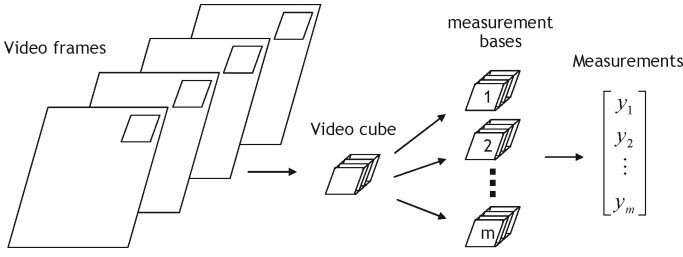


Fig. 2. Video encoding using compressive sensing.

The encoding of video is performed cube by cube on all video cubes using compressive sensing. Let $x \in R^n$ be the column vector obtained by concatenating the pixels of a video cube column by column, and then frame by frame, i.e., x is resulted from the vectorization of the 3-D video cube, where $n = p \times q \times r$ is the length of x . In general, the pixels in a video cube, especially when the frames of the cube are chosen by a motion estimate or motion compensation scheme, are highly correlated, and therefore, the vector x or the gradient of x is sparse under some basis. This means that x can be well represented and reconstructed by using compressive sensing [9]. Suppose $A \in \mathbb{R}^{m \times n}$ denotes the measurement matrix ($m \leq n$), then the m compressive measurements of x compose the vector $y \in \mathbb{R}^m$. Mathematically, we have

$$y = Ax. \quad (2.1)$$

The encoding process described above is illustrated in Fig. 2.

The measurement matrix A should be incoherent with the sparsity basis of signal x , but in general, a matrix generated by randomly selecting the entries of the columns can result in good performance [20]. A matrix built by randomly picking rows from the Fourier matrix has been proven to possess a strong recoverability [9]. In this brief paper, a randomly permuted Walsh-Hadamard matrix is used. The Walsh-Hadamard matrix is defined by a recursion by using the Kronecker product. This class of matrices has the advantage of easy implementation on hardware, and satisfactory recoverability. In practice, the measurements are made by using a fast transform, so there is no need to generate the Walsh-Hadamard matrix explicitly.

The measurements from a random matrix have equal significance in the sense that the quality of the reconstructed video does not depend on any particular, or a particular set of, measurements. The quality of the reconstructed video is a function of the number of distinct measurements received. This property is particularly desirable in wireless applications. In wireless broadcast, many measurements are transmitted, but different clients may correctly receive different subsets of measurements due to different channel conditions. A client with a higher channel capacity receives more correct measurements, and therefore, is able to reconstruct a video with a higher quality than a client with a lower channel capacity. This provides a graceful degradation in the broadcast system, and avoids the cliff effect.

This approach has an advantage even in a non-broadcast network. When the compressive measurements are used for video transmission between two stations, the receiving station

does not need to acknowledge whether a measurement is received or not. The sending station only needs to keep transmitting (distinct) measurements until the receiving station acknowledges that a sufficient number of measurements have been received, after which the transmission could stop. Since only one acknowledgement is sent, it reduces overhead and latency in the transmission.

It is also worthwhile to note that each measurement y_i has a distribution approximating the Gaussian distribution. This is because if the number of pixels in a video cube is sufficiently large and the entries of the measurement matrix are independent random variables, the central limit theorem applies to each measurement which is a sum of independent random variables. Knowledge of the distribution of the measurements helps to make more efficient quantization of measurements.

B. Reconstruction

The received measurements y may contain noise due to the poor channel condition or signal disturbance in the course of wireless transmission. We decode the video by reconstructing each of the video cubes. A video cube is recovered from the received measurements by solving either a constraint minimization problem

$$\min_x \Phi(x) \text{ such that } y = Ax, \quad (2.2)$$

for a noise-free case, or an unconstrained problem

$$\min_x \Phi(x) + \frac{\mu}{2} \|Ax - y\|_2^2, \quad (2.3)$$

for a noisy case. Here, μ is the penalty parameter to balance two parts and $\Phi(x)$ is the regularization term to handle the ill-posedness or to prohibit overfitting.

Usually, there are three types of the regularization term, namely, ℓ_0 -norm which counts the number of non-zero entries, ℓ_1 -norm which sums up the magnitude of all entries, and TV-norm which measures the discontinuities in a signal. The ℓ_0 minimization is unstable and has exponential complexity. There is extensive literature illustrating results on equivalence of ℓ_0 and ℓ_1 minimization for the problem (2.2) or (2.3), but solving ℓ_1 minimization requires much lower complexity [18, 19]. However, when x is the vector formed from the pixels of a video cube, it is not obvious in which basis x is sparse, and furthermore, in which basis, x has the most sparseness.

The total variation regularization has been widely, and successfully, applied to image processing [15, 17, 21, 23]. It is capable of detecting edges and sharp changes in a signal or 2-D image. It has been shown recently that TV transform is equivalent to wavelet transform in image restoration [29]. We propose a new regularization based on total variation and suitable for the video cube reconstruction.

To define the regularization term, we reshape the vector x into a 2-D matrix. Each column of x is formed by pixels of a frame, and the rows of x are formed by the frames in the video cube in the temporal direction. The new regularization term calculates the spatial TV of the time domain DCT coefficients of current cube, i.e.,

$$\Phi(x) = TV_s \cdot x \cdot DCT_t. \quad (2.4)$$

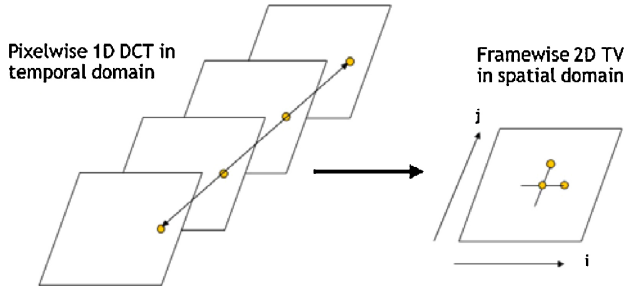


Fig. 3. Illustration of TV-DCT regularization.

Specifically, each column of $TV_s \cdot Z$ is the framewise 2-D total variation of 3-D cube Z defined as

$$TV_s \cdot Z = \sum_{i,j,k} \sqrt{(Z_{i+1,j,k} - Z_{i,j,k})^2 + (Z_{i,j+1,k} - Z_{i,j,k})^2},$$

for isotropic total variation, or

$$TV_s \cdot Z = \sum_{i,j,k} (|Z_{i+1,j,k} - Z_{i,j,k}| + |Z_{i,j+1,k} - Z_{i,j,k}|),$$

for anisotropic total variation. For vector z , generated by stacking columns of Z , $TV_s \cdot z$ is defined to be equal to $TV_s \cdot Z$, i.e.,

$$TV_s \cdot z = \sum_{l=1}^n \|D_l z\|. \quad (2.5)$$

In (2.5), $D_l z \in \mathbb{R}^2$ is the discrete gradient of Z at position l , given by

$$D_l z = (Z_{i+1,j,k} - Z_{i,j,k}, Z_{i,j+1,k} - Z_{i,j,k})^T, \quad (2.6)$$

where index l is the 1D correspondence of the 3-D index (i, j, k) . Also in (2.5) $\|\cdot\|$ can be either the ℓ_1 -norm which corresponds to the anisotropic TV or the ℓ_2 -norm which corresponds to the isotropic TV.

In (2.4), $x \cdot DCT_t$ represents the temporal discrete cosine transform of each row of x . Each column of $x \cdot DCT_t$ consists of DCT coefficients of a particular frequency. In short, using the TV-DCT regularization as the objective function minimizes the spatial total variation of the frequency components in time. The new regularization defined above is illustrated in Fig. 3.

In Fig. 3, the left hand side illustrates a source video cube with 4 frames. For each spatial location, pixels in the same spatial location from all frames are used in a 1D DCT transform. After the temporal DCT transform, each frame consists of DCT coefficients of a particular frequency. On the right hand side of Fig. 3, the DCT coefficients from each frame are used to perform the 2-D spatial TV calculation of equation (2.5).

III. EXTENDED TVAL3 ALGORITHM

In this section, we develop a new algorithm which is extended from TVAL3 [17] for the solution of problems (2.2) and (2.3). TVAL3 is a TV minimization algorithm based on the augmented Lagrangian method [22] and alternating minimization, which is widely used to solve 1D signal and 2-D image reconstruction and denoising problems. The idea

of alternating minimization was introduced to solve a series of TV minimization problems including deconvolution, denoising, and image reconstruction by Zhang *et al.* [23].

At each outer iteration of TVAL3, the algorithm minimizes the corresponding augmented Lagrangian function by solving two subproblems alternately. One subproblem is separable and has a closed-form solution while the other subproblem is quadratic and can be approximated by one-step steepest descent with an aggressive step length. We can employ similar strategy to solve both (2.2) and (2.3). Since the derivation of the algorithm is almost the same for both problems, in the following, we will present the description of the algorithm for (2.2) in details. The algorithm for (2.3) is similar.

Let $T_i \in \mathbb{R}^{2 \times n}$ denote the linear transformation satisfying $T_i x = D_i(x \cdot DCT_t)$ for any i , $1 \leq i \leq n$, where the operator D_i is defined in (2.6). Then the minimization problem (2.2) becomes

$$\min_x \sum_{i=1}^n \|T_i x\| \quad \text{such that } y = Ax. \quad (3.1)$$

By introducing a series of new variables $w_i = T_i x$, $1 \leq i \leq n$, we consider a variant of (3.1)

$$\min_{x, w_i} \sum_{i=1}^n \|w_i\| \quad \text{s.t. } y = Ax \quad \text{and } w_i = T_i x, \text{ for all } i. \quad (3.2)$$

Its corresponding augmented Lagrangian function is

$$L_A(x, w_i) = \sum_{i=1}^n (\|w_i\| - v_i^T (T_i x - w_i) + \frac{\beta_i}{2} \|T_i x - w_i\|_2^2) - w_i \|_2^2 - \lambda^T (Ax - y) + \frac{\mu}{2} \|Ax - y\|_2^2. \quad (3.3)$$

Let x^* and w_i^* denote the true minimizers of $L_A(x, w_i)$. The augmented Lagrangian method minimizes (3.3) recursively and then updates multipliers v_i and λ at the end of each iteration as follows:

$$\begin{cases} \tilde{v}_i = v_i - \beta_i (T_i x^* - w_i^*) & \text{for all } i, \\ \tilde{\lambda} = \lambda - \mu (Ax^* - y). \end{cases} \quad (3.4)$$

To develop an efficient algorithm which can handle large-scale data sets such as video streams, it is essential to minimize $L_A(x, w_i)$ at a low complexity. An iterative method based on the alternating minimization is employed to minimize $L_A(x, w_i)$ and it is described in the following.

For a given x , minimizing $L_A(x, w_i)$ is equivalent to

$$\min_{w_i} \sum_{i=1}^n (\|w_i\| - v_i^T (T_i x - w_i) + \frac{\beta_i}{2} \|T_i x - w_i\|_2^2). \quad (3.5)$$

This problem is separable with respect to each w_i and has a closed-form solution [17]:

$$\tilde{w}_i = \begin{cases} \max \left\{ |T_i x - \frac{v_i}{\beta_i}| - \frac{1}{\beta_i} \right\} \text{sgn}(T_i x - \frac{v_i}{\beta_i}), \\ \quad \text{for anisotropic TV,} \\ \max \left\{ \|T_i x - \frac{v_i}{\beta_i}\|_2 - \frac{1}{\beta_i} \right\} \frac{(T_i x - v_i / \beta_i)}{\|T_i x - v_i / \beta_i\|_2}, \\ \quad \text{for isotropic TV.} \end{cases} \quad (3.6)$$

Formula (3.6) can be derived by using 1D and 2-D shrinkage.

Algorithm 1 Alternating minimization

Initialize Starting Points and Parameters. In Particular, *Initialize* x and w_i with Initial Guess
while not converge, **do**
 for the given x , compute \tilde{w}_i according to (3.6).
 update $w_i \leftarrow \tilde{w}_i$
 for the given w_i , compute \tilde{x} according to (3.8), by taking one step of steepest descent.
 update $x \leftarrow \tilde{x}$
end do

On the other hand, for a given w_i , minimizing $L_A(x, w_i)$ is equivalent to

$$\min_x Q(x) = \min_x \left(\sum_{i=1}^n \left(-v_i^T (T_i x - w_i) + \frac{\beta_i}{2} \|T_i x - w_i\|_2^2 \right) - \lambda^T (Ax - y) + \frac{\mu}{2} \|Ax - y\|_2^2 \right). \quad (3.7)$$

Clearly, $Q(x)$ is a quadratic function and can be minimized by various iterative methods. However, these methods may be too costly for large-scale data sets and unnecessary since our ultimate goal is to solve (3.1) or (3.2) instead of (3.7). Therefore, a good approximation of the true minimizer of (3.7) should be good enough to guarantee the convergence. Specifically, we take one step of the steepest descent with an aggressive step length and accept it as the approximate solution of (3.7):

$$\tilde{x} = x - \alpha d(x). \quad (3.8)$$

Here, α denotes the step length and $d(x)$ denotes the gradient of quadratic function $Q(x)$ at the previous iteration x . As proposed in [17], the step length can be iteratively determined by a nonmonotone line search [26], starting from the initial guess given by the Barzilai-Borwein method [16]. The step length computed like this has been demonstrated to be effective and efficient in practice.

Therefore, the algorithm to minimize the augmented Lagrangian function (3.3) can be described as follows:

The above algorithm minimizes the Lagrangian function (3.3) for fixed values of the multipliers v_i and λ . After Algorithm 1 is performed, these multipliers are updated by using equations (3.4) with x^* and w_i^* replaced by x and w_i computed from Algorithm 1, respectively. After the update of the multipliers, Algorithm 1 is applied again and the multipliers are further updated in another iteration. This defines the iterative process of solving the minimization problem (3.1), and equivalently, (2.2). In order to speed up the convergence of the iteration, a continuation method can be used in which the penalty parameters β_i and μ may be updated from iteration to iteration. For example, at the start of the iteration, the initial values for the penalty parameters are chosen to be much smaller than the target values of the penalty parameters. These values are then increased (or kept the same, but non-decreasing) at each iteration until the target values of β_i and μ are reached. In summary, the overall algorithm is provided in Algorithms 1 and 2.

The framework of Algorithm 2 described above can be also used to solve a larger range of minimization problems (2.2)

Algorithm 2 Parameter update

Initialize Starting Points, Multipliers, and Penalty Parameters
while not converge, **do**
 find the minimizers of the augmented Lagrangian function (3.3) by means of Algorithm 1, starting from the previous iterates.
 update the multipliers according to (3.4).
 choose the new penalty parameters.
end do

and (2.3) with different regularizations, such as 3-D TV, ℓ_1 under sparsity basis, etc.

IV. SIMULATION

A. Simulation Using Compressive Sensing Methods

Simulations are performed on two source video clips, *Container* and *News*. They are both CIF resolution (352×288) with a frame rate of 30 f/s. In the simulations, a source video is encoded with compressive measurements as described in Section II.A. A permuted Walsh-Hadamard matrix is used as the measurement matrix. Each cube consists of 8 full CIF frames, i.e., the total number of pixels in each video cube is $n = 352 \times 288 \times 8 = 811008$. After coding, a number of randomly selected measurements, m , is used for reconstruction. The random selection of m measurements simulates a channel of capacity m/n . That is, it is assumed that a total of n measurements are transmitted, and only m of transmitted measurements are correctly received and used in reconstruction. The PSNR of the reconstructed video is calculated as a function of the percentage of the measurements received (m/n).

For each set of correctly received measurements, four different regularizations are used to reconstruct the video including the one proposed in this brief paper, then the PSNRs of reconstructed videos are compared. To fairly compare the results, reconstruction algorithms for the different models were all implemented similarly to what is described in Section III. The four regularization terms are described below.

2-D TV + pointwise DCT (TV-DCT): This is the method proposed in this brief paper and described in Sections II and III.

2-D TV: This is the method minimizing the sum of 2-D TV of every frame. In other words, the video cube is treated as individual frames, and no temporal relations are explored. Mathematically, let X denote the original 3-D cube of x before vectorization, and 2-D TV regularization is defined as

$$\Phi(x) = \sum_{i,j,k} \left(|X_{i+1,j,k} - X_{i,j,k}| + |X_{i,j+1,k} - X_{i,j,k}| \right).$$

3-D TV: This is the method minimizing the 3-D TV of the whole cube, assuming the sparsity of gradients in both spatial and temporal directions. Mathematically, under the same notation, 3-D TV regularization is defined as

$$\Phi(x) = \sum_{i,j,k} \left(|X_{i+1,j,k} - X_{i,j,k}| + |X_{i,j+1,k} - X_{i,j,k}| + |X_{i,j,k+1} - X_{i,j,k}| \right).$$

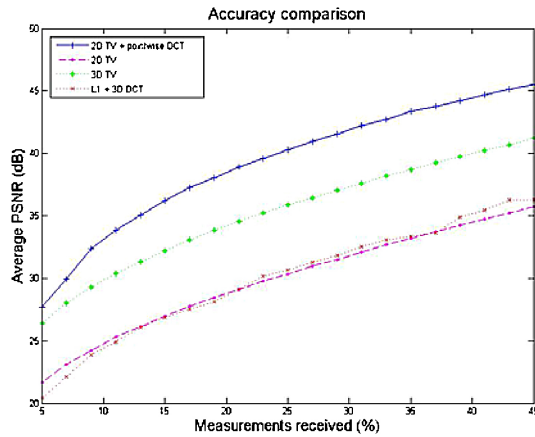


Fig. 4. PSNR for video container.

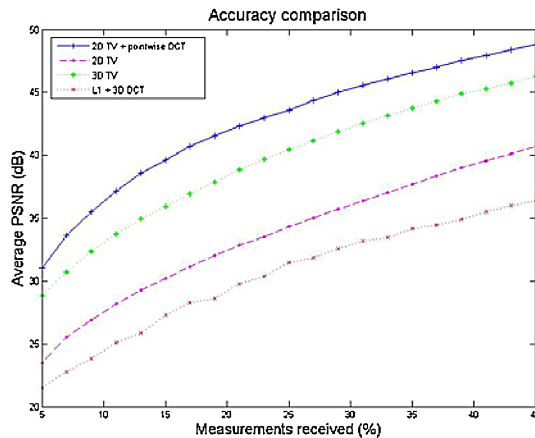


Fig. 5. PSNR for video news.

$\ell_1 + 3\text{-D DCT}$: This is the method minimizing the ℓ_1 -norm of the 3-D DCT coefficients of the source cube. That is, a 3-D DCT transform is performed on the video cube, and the ℓ_1 -norm of the resulting coefficients is minimized. Mathematically, let $DCT_3(x)$ denote the vectorization of 3-D DCT of the cube X , ℓ_1 regularization under 3-D DCT basis is defined as $\Phi(x) = \|DCT_3(x)\|_1$.

In all four methods, the same permuted Walsh-Hadamard matrix is used as the sensing matrix, and the received measurements used for reconstruction are randomly chosen. The accuracy of reconstruction, as measured by PSNR of the reconstructed frames with the original frames, is reported in Figs. 4–5.

Two observations can be made from these results. First the compressive sampling methods, 2-D TV+pointwise DCT, 2-D TV, 3-D TV, and $\ell_1+3\text{D DCT}$ all have the scalability property desired for wireless transmission, namely, the PSNR of the reconstructed video increases progressively with the number of measurements received. This avoids the cliff effect seen with video coding such MPEG2 or H.264.

The second observation is that the TV minimization based methods are superior to the ℓ_1 minimization of 3-D DCT. Among all methods evaluated, the method proposed in this brief paper is clearly better than the other three.

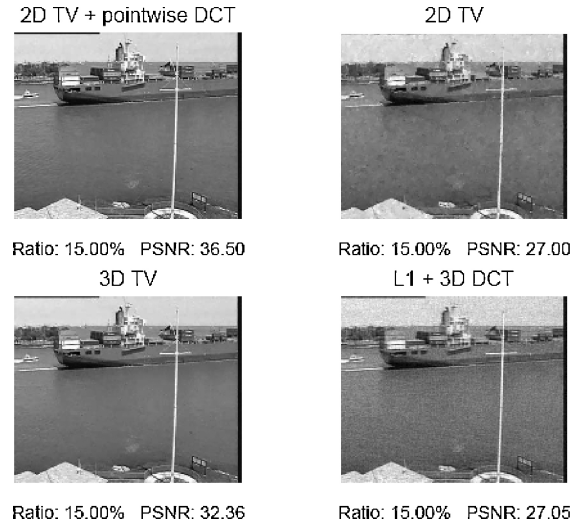


Fig. 6. Frame 4 of recovered video container.

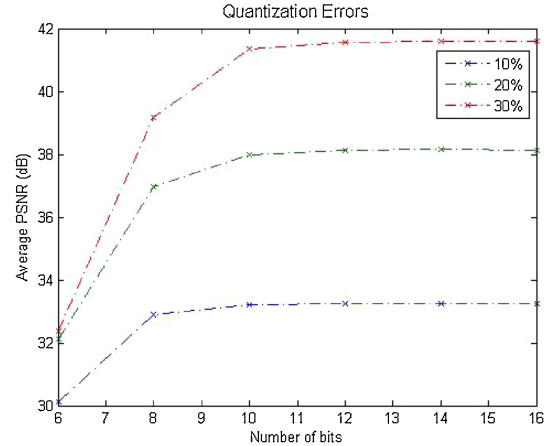


Fig. 7. Quantization impact for container video.

A typical frame in the recovered video for the *Container* clip for each of the four tested methods is shown in Fig. 6.

Fig. 6 shows the quality of the reconstructed frame 4 by using different methods. The same number of measurements is used in all methods, and the quality, in terms of PSNR, of the reconstructed video frame for each method is reported. As is evident, the method of this brief paper, 2DTV+pointwise DCT, has the best PSNR.

Next, we present results regarding the impact of quantization on the performance of the reconstruction using the proposed method. Each received measurement is quantized to a fixed number of bits before they are used for the reconstruction. The PSNR of the reconstructed video is measured as a function of the number of bits in the quantization. The results are reported for different percentages of measurements used in reconstruction, and they are shown in Figs. 7 and 8.

It can be concluded from Figs. 7–8 that the reconstruction is not sensitive to the number of bits in the quantization when the number of bits is above 10. For a given quantization level (a given number of bits per measurement), the quality of the video can still be improved by increasing the number of measurements used in reconstruction.

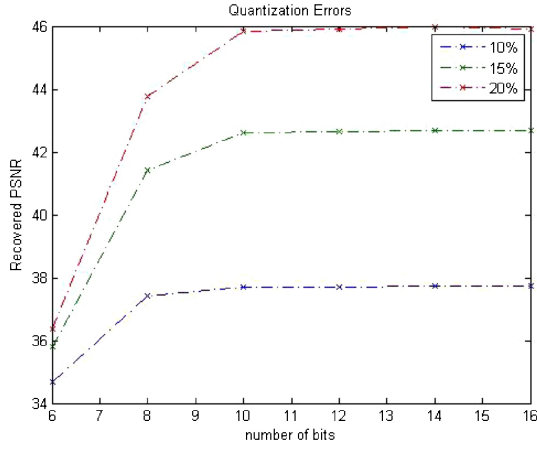


Fig. 8. Quantization impact for news video.

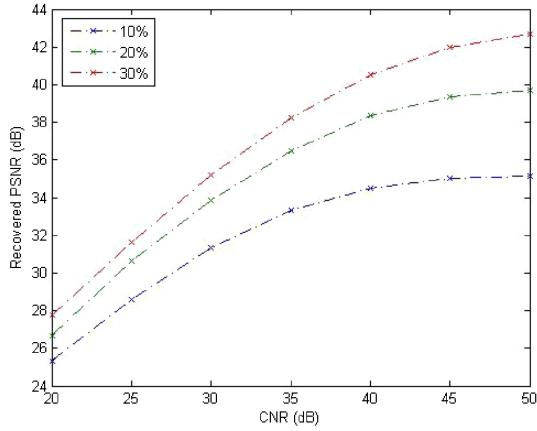


Fig. 9. Impact of additive Gaussian noise for container video.

Finally, simulations are performed when channel noise is present. In the simulation, the selected measurements are injected with Gaussian noise, and the resulting noisy measurements are used for reconstruction. In other words, the reconstruction is performed by minimizing (2.3) with y replaced by $\hat{y} = y + n$, where \hat{y} is the received measurements with additive Gaussian noise n . The PSNR of the reconstructed video (by using \hat{y}) as a function of the carrier to noise level, CNR, is measured. The results are reported for different percentages of measurements used in reconstruction, and they are shown in Figs. 9–10.

It is demonstrated in Figs. 9–10 that our reconstruction algorithm is reliable when noise is present in the measurements. Furthermore, for a given amount of noise present in the measurements, the quality of video can be improved by using more measurements. This is important for mobile broadcast. The measurements may be broadcast in a method proposed in [8] in which one transmission serves all clients. The amount of channel noise is the sole determination of the quality of the reconstructed video. This further demonstrates the coding method of this brief paper is scalable with the transmission channel.

B. Comparison with H.264

We now present simulation results for H.264 and compare them with the compressive sensing results. In the simulation, we use a broadcast channel model.

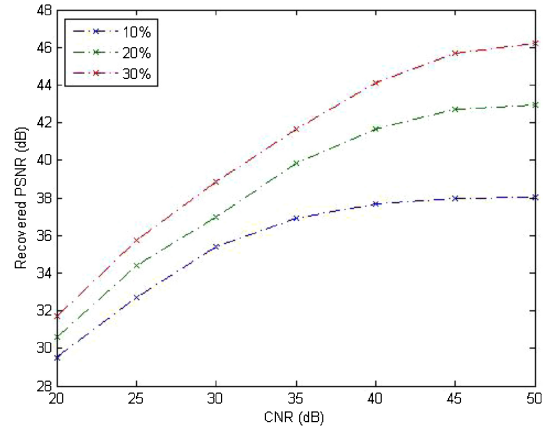


Fig. 10. Impact of additive gaussian noise for news video.

Broadcast Erasure Channel Model We consider a broadcast erasure channel (BEC) [30] without feedback. In a BEC model, data is transmitted by a single transmitter, and received by multiple clients. Each broadcast packet is either received correctly by a client or is lost (dropped), in which case an erasure occurs for that client. Each client has its own channel rate R , which is the rate of correctly received packets. Different clients may have different channel rate. BEC models have been used to analyze wireless networks such as 802.11, and broadcast systems such as DVB-SH if an uncorrectable packet is dropped and not forwarded to the upper layer.

H.264 Encoding/Decoding Video sequences are encoded as H.264 streams using a publicly available H.264 codec, FFmpeg [31]. The default encoding parameters are used but the quantization levels are chosen such that the encoded video has a desired PSNR (design PSNR). Specifically, for the News clip, the encoded stream has a bit rate of 440kbps with a design PSNR of 43dB, and for the Container clip, the encoded stream has a bit rate of 490kbps with a design PSNR of 40dB. The rate at which the encoded video achieves the design PSNR is defined as the design rate R_d .

After encoding, the H.264 bit streams are modified to emulate the broadcast erasure channel. A number of bits are dropped from the encoded bit streams, and for a client channel rate of R , the percentage of the encoded bits correctly received and used in the decoding is equal to $\frac{R}{R_d}$.

Next, the modified bit streams are decoded using the FFmpeg decoder [31] with error concealment turned off. The decoded video is compared with the original video to calculate PSNR.

Compressive sensing (CS) The compressive sensing (CS) method used is the TV-DCT method described previously. Measurements are made by a randomly permuted Walsh-Hadamard matrix. The measurements are then quantized to 10 bits. In BEC model, a measurement is either correctly received or dropped. The client channel rate R represents the number of correctly received measurements per second. The PSNR is calculated by comparing the reconstructed video with the original video.

The results from H.264 and CS are compared in Fig. 11. The PSNR values of the reconstructed video is plotted against the

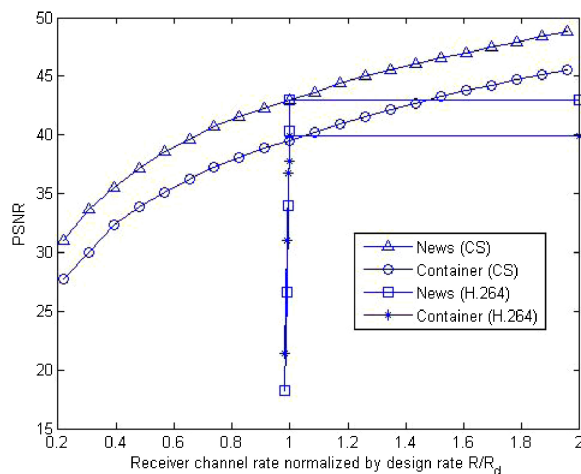


Fig. 11. PSNR versus normalized client channel rate.

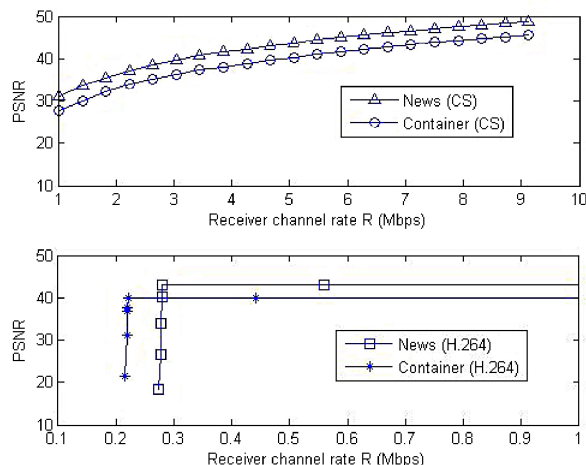


Fig. 12. PSNR versus client channel rate.

normalized client channel rate $\frac{R}{R_d}$. The figure clearly shows that standard H.264 codecs exhibit a sharp drop in video quality when the client channel rate is below the design rate R_d . Note that the PSNR values for the H.264 streams are less than 20dB when client channel rate is only 2% below the design rate, while the video quality from CS method degrades gracefully. We point out that a PSNR value below 20dB represents severely-degraded video which is non-viewable in practice. In addition, for clients that have higher channel rate than the design rate, the video quality can never get better in H.264 than the design PSNR. Essentially, the H.264 streams are optimized for channel rates in a very narrow range around the design rate.

Fig. 12 replots the same data as Fig. 11 using absolute channel rates. The top plot shows the PSNR values from CS and the bottom plot shows those from H.264. To achieve the same video quality, CS currently requires higher channel rate.

However, this is a somewhat unfair comparison as H.264 has the advantage of both entropy encoding and motion compensation, while CS does not currently use either. Entropy encoding can be easily introduced in CS, and it should have

similar impact to CS as it has to H.264. Motion compensation may be introduced during the reconstruction, for example, by an iterative process in which no motion is used in the first pass in the reconstruction, and after the video is reconstructed, a motion estimate is performed to define a trajectory for DCT operation in the next iteration of the reconstruction. Although motion compensation can be used in CS method, it is not clear what its impact is as compared to the impact of motion compensation in H.264. Further study is needed to investigate the true compression ratio of CS method when motion compensation is incorporated.

V. CONCLUSION

A framework for video coding using compressive sensing is proposed. In this framework, the source video is divided into video cubes, and measurements are made using random matrices. Video is reconstructed by performing the minimization of the framewise total variation of the coefficients from the pixelwise temporal DCT transform. Aiming at the new TV-DCT regularization, an efficient reconstruction algorithm based on TVAL3 is described. This new framework results in a video coding that is suitable for wireless transmission of video. Simulations demonstrate that the proposed method produces videos having higher PSNR than other compressive sampling recovery methods.

The work in this brief paper is mainly concerned with compressive video coding that is scalable with channel capacity, which is of particular interest in wireless broadcast situations since mobile clients may experience vastly different channel conditions. Further investigation is needed for this framework to become practical in wireless applications. The compression ratio can clearly be improved through the use of entropy encoding, and various schemes to use motion vectors during coding or solely during reconstruction show promise.

ACKNOWLEDGMENT

The authors would like to thank K. Matthews and G. Huang, Alcatel-Lucent, and W. Yin, Rice University, Houston, TX, for insightful discussions. They would also like to thank the reviewers for their valuable comments that improved this brief paper.

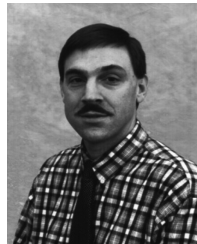
REFERENCES

- [1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
- [2] D. S. Taubman and M. W. Marcellin, *JPEG 2000 Image Compression Fundamentals, Standards and Practice*. The Netherlands: Kluwer Academic, 2001.
- [3] H. Jiang and P. Wilford, "A hierarchical modulation for upgrading digital broadcast systems," *IEEE Trans. Broadcasting*, vol. 51, no. 2, pp. 223–229, Jun. 2005.
- [4] A. M. C. Correia, J. C. M. Silva, N. M. B. Souto, L. A. C. Silva, A. B. Boal, and A. B. Soares, "Multi-resolution broadcast/multicast systems for MBMS," *IEEE Trans. Broadcasting*, vol. 53, no. 1, pp. 224–234, Mar. 2007.
- [5] H. Sohn, H. Yoo, W. De Neve, C. S. Kim, and Y. M. Ro, "Full-reference video quality metric for fully scalable and mobile SVC content," *IEEE Trans. Broadcasting*, vol. 56, no. 3, pp. 269–280, Sep. 2010.

- [6] L. Zhou, B. Zheng, A. Wei, B. Geller, and J. Cui, "A robust resolution-enhancement scheme for video transmission over mobile ad-hoc networks," *IEEE Trans. Broadcasting*, vol. 54, no. 2, pp. 312–321, Jun. 2008.
- [7] J. Xu, X. Shen, J. W. Mark, and J. Cai, "Adaptive transmission of multilayered video over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, pp. 2305–2314, Jun. 2007.
- [8] S. Jakubczak, H. Rahul, and D. Katabi, "SoftCast: One video to serve all wireless receivers," *Comput. Sci. Artif. Intell. Lab., Tech. Rep., MIT-CSAIL-TR-2009-005*, Feb. 2009.
- [9] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [10] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. PCS*, Apr. 2006.
- [11] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive video sampling," in *Proc. EUSIPCO*, Aug. 2008.
- [12] T. Do, Y. Chen, D. Nguyen, N. Nguyen, L. Gan, and T. Tran, "Distributed compressed video sensing," in *Proc. 16th IEEE ICIP*, Nov. 2009, pp. 1393–1396.
- [13] J. Prades-Nebot, Y. Ma, and T. Huang, "Distributed video coding using compressive sampling," in *Proc. PCS*, 2009, pp. 1–4.
- [14] S. Pudlewski and T. Melodia, "On the performance of compressive video streaming for wireless multimedia sensor networks," in *Proc. IEEE ICC*, May 2010.
- [15] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14–20, Mar. 2008.
- [16] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.
- [17] C. Li, "An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing," M.S. thesis, *Comput. Appl. Math., Rice Univ., Houston, TX*, 2009.
- [18] D. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inform.*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001.
- [19] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [20] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [21] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, nos. 1–4, pp. 259–268, 1992.
- [22] M. R. Hestenes, "Multiplier and gradient methods," in *Computing Methods in Optimization Problems*, 2nd ed., L. A. Zadeh, L. W. Neustadt, and A. V. Balakrishnan, Eds. New York, USA: Academic Press, 1969, pp. 143–163.
- [23] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, no. 4, pp. 248–272, 2008.
- [24] B. Katz, S. Greenberg, N. Yarkoni, N. Blaunstein, and R. Giladi, "New error-resilient scheme based on FMO and dynamic redundant slices allocation for wireless video transmission," *IEEE Trans. Broadcasting*, vol. 53, no. 1, pp. 308–319, Mar. 2007.
- [25] D. A. Adjeroh and S. D. Sawant, "Error-resilient transmission for 3-D DCT coded video," *IEEE Trans. Broadcasting*, vol. 55, no. 2, pp. 178–189, Jun. 2009.
- [26] H. Zhang and W. W. Hager, "A nonmonotone line search technique and its application to unconstrained optimization," *SIAM J. Optim.*, vol. 14, no. 4, pp. 1043–1056, 2004.
- [27] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic MRI," *Magn. Resonance Med.*, vol. 59, no. 2, pp. 365–373, 2008.
- [28] H. Jung, K. Sung, K. Nayak, E. Kim, and J. Ye, "k-t FOCUSS: A general compressed sensing framework for high resolution dynamic MRI," *Magn. Resonance Med.*, vol. 61, no. 1, pp. 103–116, Jan. 2009.
- [29] J. Cai, B. Dong, S. Osher, and Z. Shen, "Image restoration: Total variation, wavelet frames, and beyond," *J. Amer. Math. Soc.*, vol. 25, no. 4, pp. 1033–1089, 2012.
- [30] M. Gatzianas, L. Georgiadis, and L. Tassioulas, "Capacity-achieving encoding for the broadcast erasure channel with multiple users," in *Proc. IEEE ISIT*, 2011, pp. 1549–1553.
- [31] *FFmpeg Source Code*. (2012, Jun. 8) [Online]. Available: <http://www.ffmpeg.org/>



pression.



MPEG2 encoder. He has made fundamental contributions in the high-speed optical transmission area. Currently, he is leading a department working on next-generation video transport systems, hybrid satellite-terrestrial networks, and high-speed mobility networks. His research interest was communication theory and predictive coding.



ing from various applications, including image and data processing.



Chengbo Li received the B.S. degree in mathematics from Nanjing University, Nanjing, China, in 2007, and the Ph.D. degree in computational and applied mathematics from Rice University, Houston, TX, USA, in 2011.

He is currently a Research Geophysicist with the Technology-Seismic Technology Organization, ConocoPhillips, Houston. His current research interests include developing and studying numerical algorithms for ill-posed inverse problems, compressive sensing, hyperspectral imaging, and video com-

Hong Jiang received the B.S. degree from Southwestern Jiaotong University, Chengdu, China, the M.Math. degree from the University of Waterloo, Waterloo, ON, Canada, and the Ph.D. degree from the University of Alberta, Edmonton, Canada.

He is currently with Bell Labs, Alcatel-Lucent Technologies, where he conducts research on digital communications, image and video processing. He has authored more than 50 technical papers in scientific and engineering journals. He holds over 40 U.S. patents in digital communications.

Paul Wilford received the B.S. and M.S. degrees in electrical engineering from Cornell University, Ithaca, NY, USA, in 1978 and 1979, respectively.

He is a Fellow with Bell Labs, and the Director of the Video and Mobile Packet Networking Research Department. He has made extensive contributions in the development of digital video processing and multimedia transport technology. He was a key leader in the development of Lucent's first HDTV broadcast encoder and decoder. Under his leadership, Bell Laboratories then developed the world's first MPEG2 encoder. He has made fundamental contributions in the high-speed optical transmission area. Currently, he is leading a department working on next-generation video transport systems, hybrid satellite-terrestrial networks, and high-speed mobility networks. His research interest was communication theory and predictive coding.

Yin Zhang received the B.S. degree in environmental engineering from the Chongqing Institute of Architecture and Engineering, Chongqing, China, in 1977, and the Ph.D. degree in applied mathematics from the State University of New York at Stony Brook, Stony Brook, NY, USA, in 1987.

He is currently a Professor with the Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA. His current research interests include developing and studying numerical algorithms for practical optimization problems arising from various applications, including image and data processing.

Mike Scheutzw received the M.S.E.E. degree from the Georgia Institute of Technology, Atlanta, GA, USA.

He has been involved with video compression research and development since 1991. His current research interests include real-time transcoding in a lossy wireless environment.